Predicting plant pathology with convolution neural network

Barış Deniz Sağlam

e155841

Problem Statement

Pathologies in apple plants cause significant economic losses.

Manual diagnosis is slow and expensive.

We develop a computer vision based solution that detect plant pathologies from the images of leaves.

Dataset

Plant Pathology 2021 (Thapa et al. 2020)

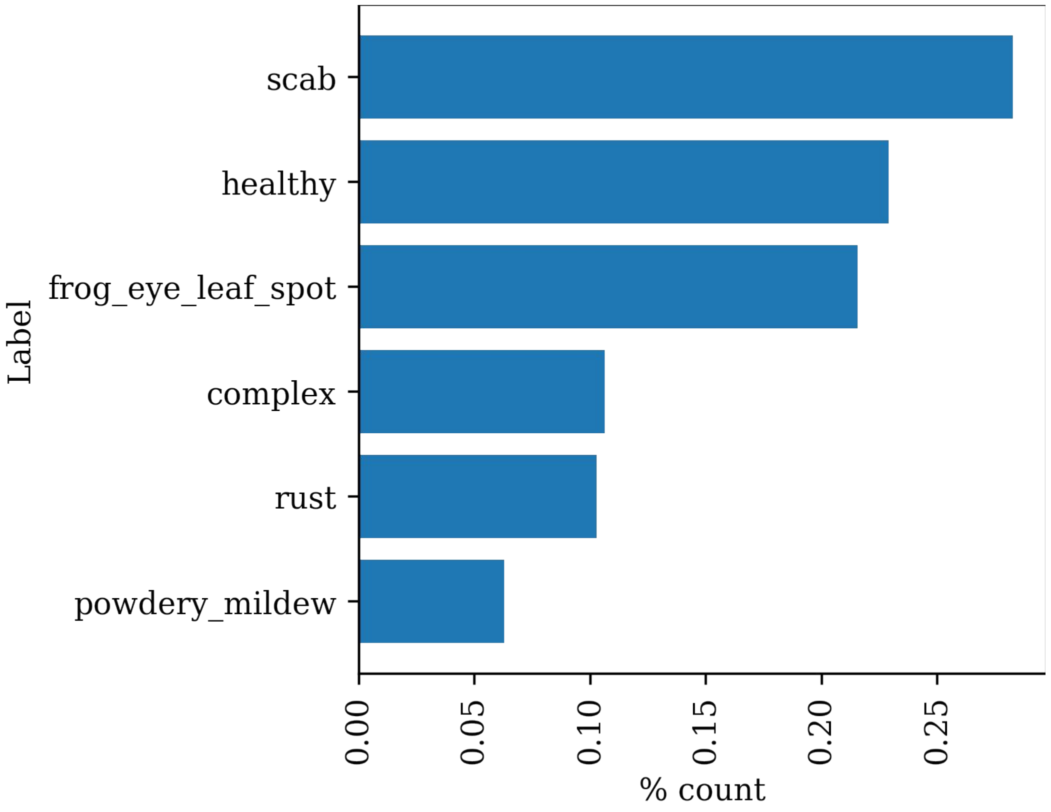
18,632 RGB images

2592 x 1728 - 5184 x 3456

Classes:

- frog-eye leaf spot
- powdery mildew
- rust
- scab
- *complex*
- *healthy*

Label distribution



Model architectures

- ResNet18, ResNet34

  K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

- EfficientNet-B2, EfficientNet-B3, EfficientNet-B5

  M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proceedings of the 36th International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

- XSE_ResNeXt50

  T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 558–567.

Loss functions

- Binary Cross Entropy Loss

$$\mathrm{BCE} = -(y\log(p) + (1-y)\log(1-p))$$

- Weighted Binary Cross Entropy Loss

complex	frog_eye_leaf_spot	powdery_mildew	rust	scab
2.6056	1.2822	4.3811	2.7743	0.9718

- Focal Loss

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

$$p_T = \begin{cases} p & \text{if y = 1} \\ 1-p & \text{otherwise} \end{cases}$$

$$\mathrm{FL}(p_t) = \alpha_t(1-p_t)^\gamma \log p_t$$

Training

fastai & PyTorch libraries

J. Howard and S. Gugger, “Fastai: A layered api for deep learning,” Information, vol. 11, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2078- 2489/11/2/108

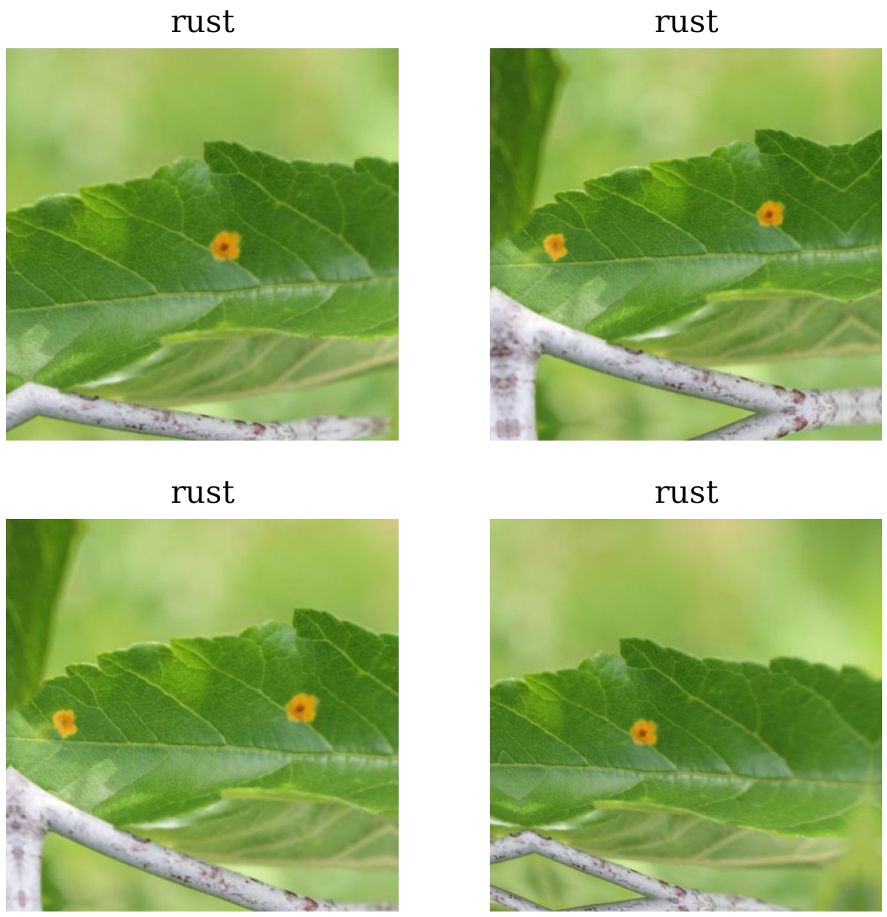
A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in Advances in Neural Information Processing Systems 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche ́-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024– 8035. [Online]. Available: http://papers.neurips.cc/paper/9015- pytorch-an imperative-style- high- performance-deep-learning-library.pdf

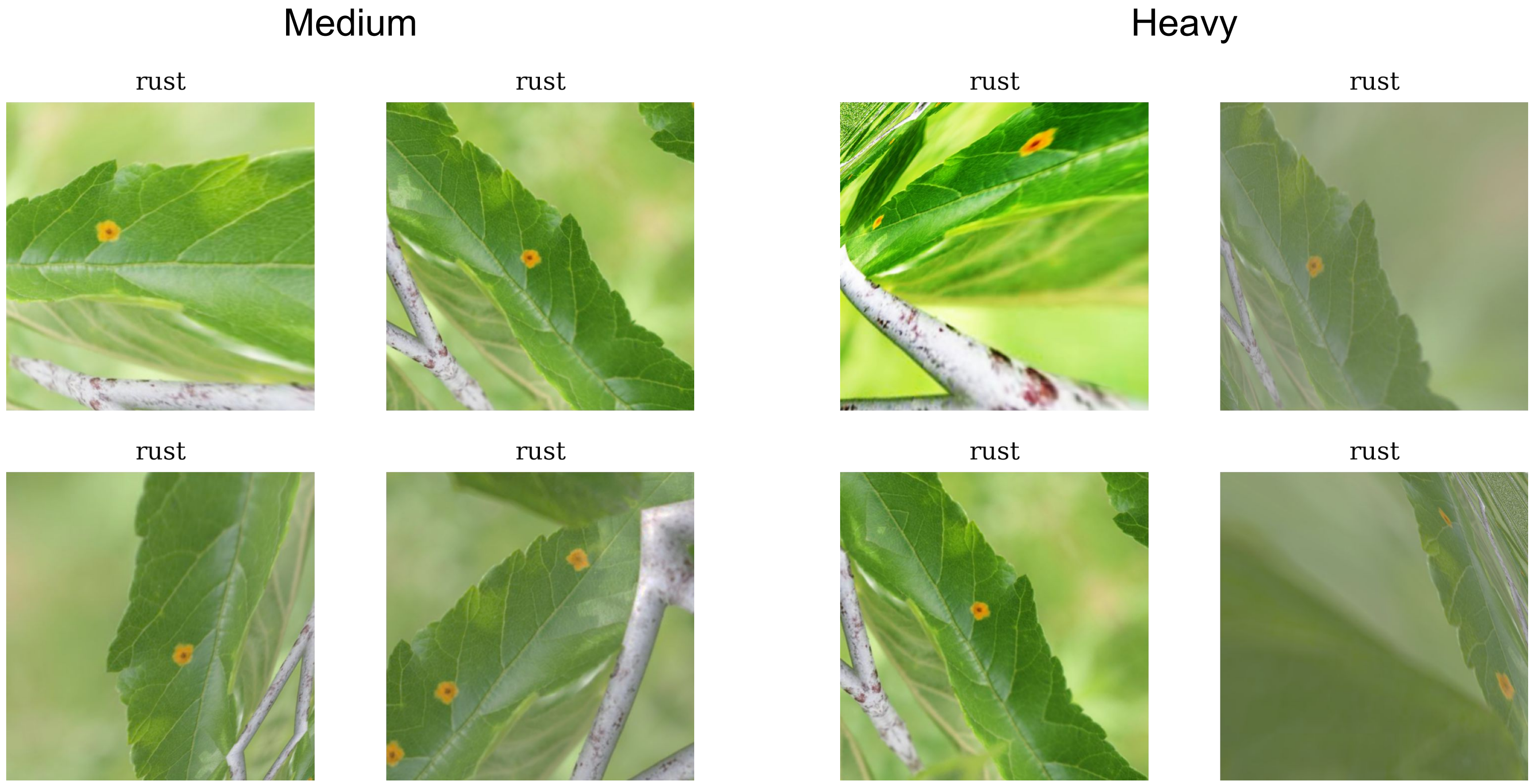
Techniques

- Image Augmentation
- Transfer Learning
- Low-precision numbers (fp16)
- Learning rate finding
- 1-cycle learning rate policy

Image augmentation

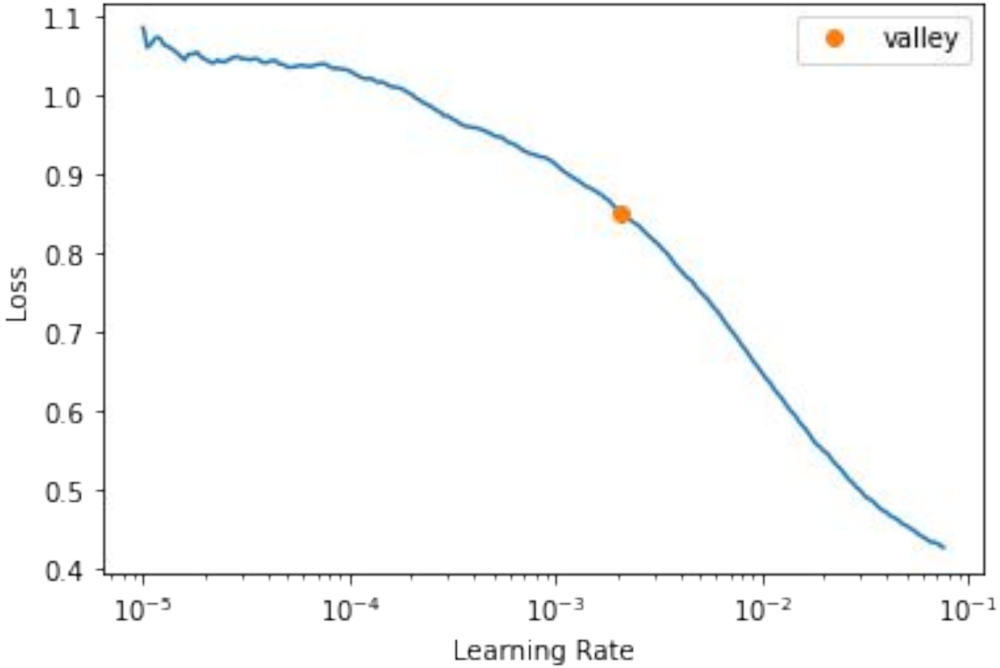
- Random Crop
- Translation
- Rotation
- Lighting
- Zoom in/out




Medium
Heavy
rust
rust
rust
rust
rust
rust
rust
rust

Learning rate finding

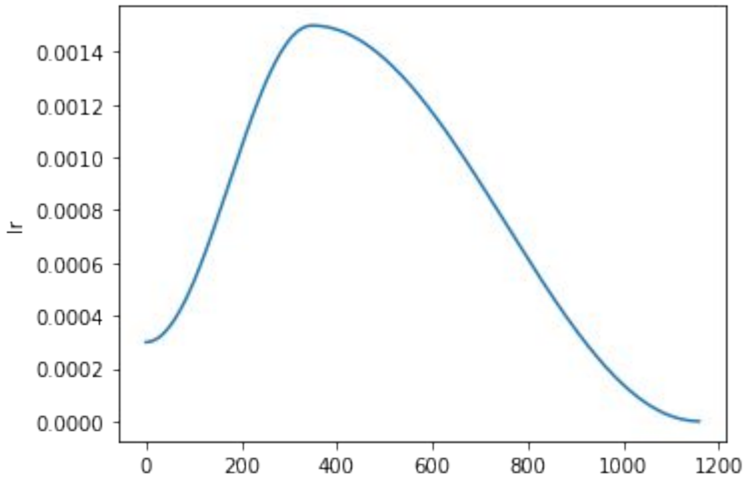
L. N. Smith, “No more pesky learning rate guessing games,” CoRR, vol. abs/1506.01186, 2015. [Online]. Available: http://arxiv.org/abs/1506.01186

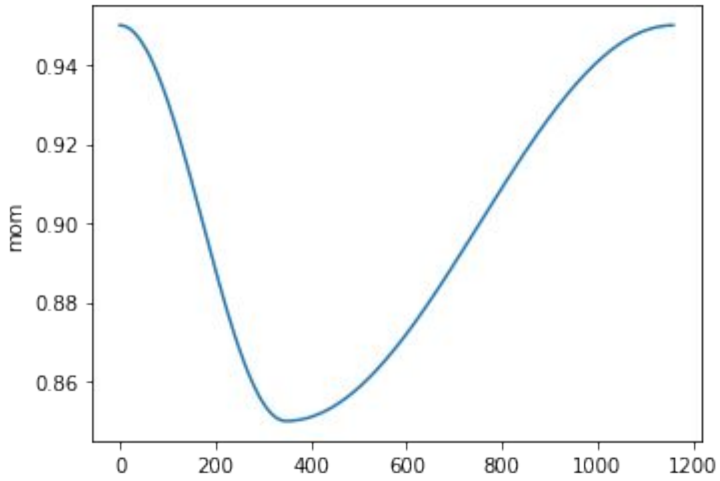


Learning Rate Schedule

1-cycle rate policy

L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay,” CoRR, vol. abs/1803.09820, 2018. [Online]. Available: http://arxiv.org/ abs/1803.09820





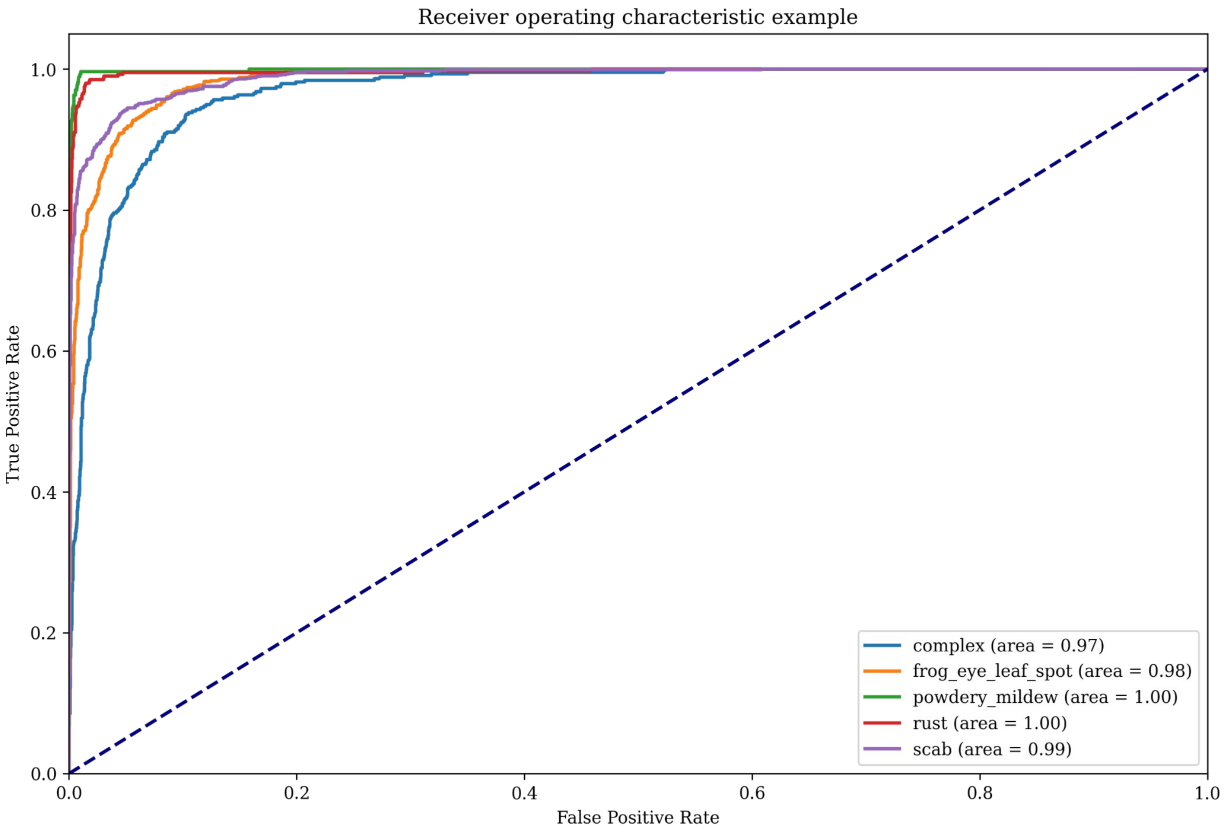
Experiment Results

Arch.	**#param [M]**	**Img size**	**Aug.**	**Loss**	**Acc.**	**F1(macro)**	**F1(samples)**
Baseline (uni.random)	0	NA	NA	NA	0.1672	0.277	0.270
ResNet18	11.7	384	None	BCE	0.963	0.888	0.675
ResNet18	11.7	384	1	BCE	**0.966**	**0.895**	**0.690**
ResNet18	11.7	384	2	BCE	0.963	0.889	0.683
ResNet18	11.7	384	1	FL	0.961	0.874	0.671
ResNet18	11.7	384	1	WBCE	0.936	0.792	0.587
ResNet34	21.8	384	2	BCE	0.965	0.891	0.681
EfficientNet-B2	9.2	384	1	BCE	0.957	0.865	0.664
EfficientNet-B3	12.3	512	2	FL	0.960	0.873	0.657
XSE-ResNext50	27.7	384	2	BCE	0.895	0.658	0.443

The experiments are tracked with Weights&Biases and the results are publicly available here.

https://wandb.ai/bdsaglam/di504-better?workspace=user-bdsaglam

Receiver Operation Characteristic Curve



Prediction samples

Ground Truth: ['scab']
Prediction: ['scab']

Ground Truth: ['frog_eye_leaf_spot']
Prediction: ['frog_eye_leaf_spot']

Ground Truth: ['frog_eye_leaf_spot', 'rust']
Prediction: ['complex', 'rust']

Ground Truth: ['scab']
Prediction: ['scab']

Ground Truth: ['complex', 'powdery_mildew']
Prediction: ['complex']

Ground Truth: ['complex']
Prediction: ['complex']

Ground Truth: ['scab']
Prediction: ['scab']

Ground Truth: ['complex', 'powdery_mildew']
Prediction: ['complex', 'powdery_mildew']